KARPOV.COURSES >>>
КОНСПЕКТ

> Конспект > 2 урок > Метрики в задаче ранжирования

> Оглавление

> Что измерять в ранжировании?
> Оценка качества ранжирования
    Пример
> Precision, Recall, Fb-мера и PR-кривая
    Почему не accuracy?
    Precision
    Recall
    Fb-мера
    PR-кривая
        Пример
> Average Precision
    Пример
    Mean Average Precision
> Переход к многоуровневой задаче, Gain
    Пример
    Normalized DCG
> PFound (Yandex)
> Исторические метрики
    MRR
        Пример
    Kendall rank correlation coefficient (Kendall's τ)
> Резюме

> Что измерять в ранжировании?

Качество/точность — *насколько аккуратна система ранжирования?*

Измеряем возможности системы ранжировать релевантные документы выше нерелевантных.

Эффективность — *насколько быстро выдается ответ? Какое количество ресурсов необходимо для формирования ответа?*

Измеряем затраты на память и время формирования ответа.

Удобство использования — *насколько полезна система для решения задач?*

Пользовательские ощущения, UX.

> Оценка качества ранжирования

Методология оценки Кранфилда (Cranfield Evaluation Methodology):

- Зафиксированный набор документов;
- Зафиксированный набор запросов;
- Оценки релевантности пар (в идеале оценки даются пользователями системы);
- Наборы должны быть репрезентативными.

Пример

Пусть у нас есть набор запросов $Q_1, Q_2, ...Q_n$ и база документов $D_1, D_2, ...D_m$.Также у нас есть оценки релевантности каждого запроса с каждым документом.

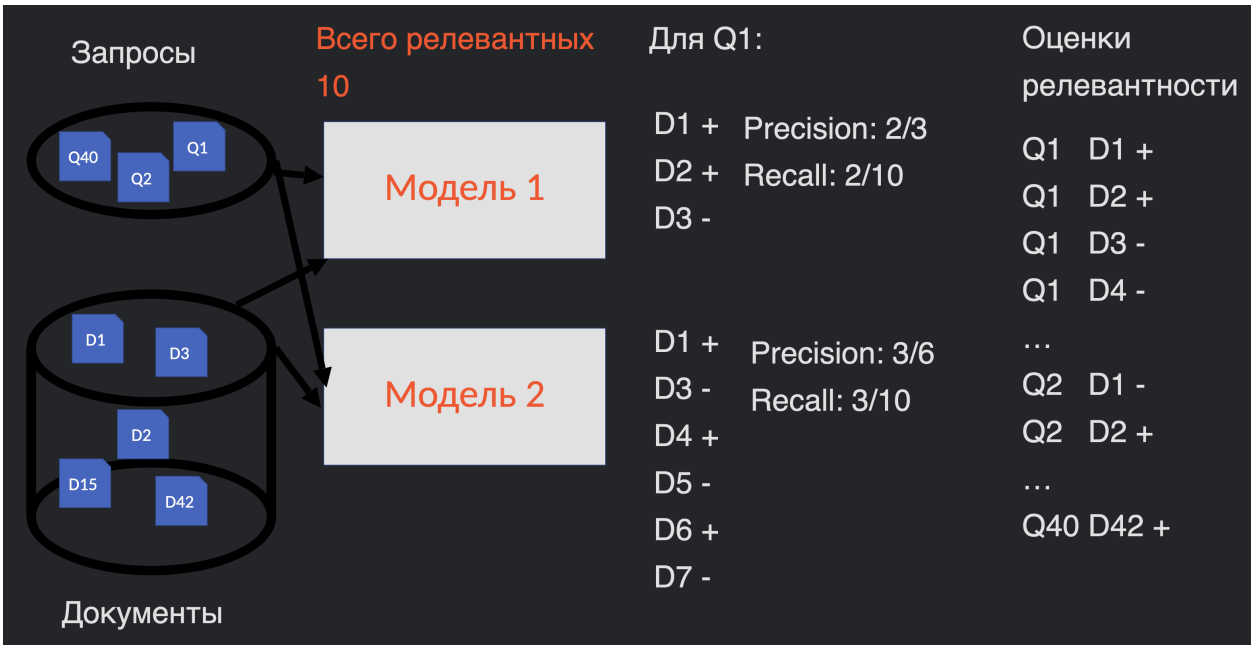
\+ *плюс*: документ соответствует запросу.

\- *минус*: документ нерелевантен.

В модель 1 и модель 2 подаём документы и запросы (рассмотрим далее Q_1), а от них получаем предсказания по релевантным документам:

- Модель 1: три документа;
- Модель 2: шесть документов.

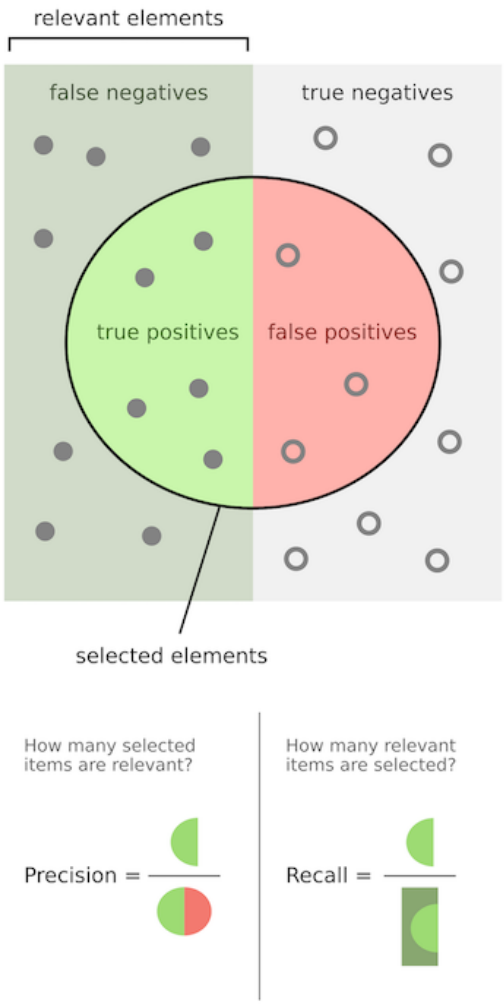
С помощью заранее заготовленной и размеченной таблицы релевантности мы можем посчитать, насколько много полезной информации в нашей выдаче.



Выбор модели зависит от решаемой задачи

> Precision, Recall, Fb-мера и PR-кривая

Метрики точности (*precision*) и полноты (*recall*) — это основные метрики в информационном поиске. Они применимы к задаче ранжирования, когда наша разметка содержит два класса: релевантно и нерелевантно.



Почему не accuracy?

Доля правильных ответов (*accuracy*) является плохой метрикой, так как практически всегда у нас невероятно сильный дисбаланс классов. Можно сделать 99.9% accuracy системой, которая практически ничего не выдает. Но что более важно — метрика получается очень нечувствительной, так как мы буквально находимся на её критическом значении, т.е. около единицы (максимума).

Метрики обычно рассчитываются по топу документов — например, по выдаче K документов. Тогда используется обозначение $metric@K$. Например, мы рассчитали точность для $K = 5$ первых документов: $Precision@5$.

Precision

Precision — доля объектов, отнесённых классификатором к положительным и действительно являющимися положительными.

`relevant documents` — релевантные документы

`retrieved documents` — выданные документы

$$precision = \frac{|\{relevant\ documents\}| \cap |\{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

Recall

Recall показывает, какую долю объектов положительного класса из всех объектов положительного класса нашёл алгоритм.

$$recall = \frac{|\{relevant\ documents\}| \cap |\{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

Fb-мера

F_β-мера — агрегированный критерий качества `precision` и `recall` , где β показывает вес точности в метрике.

F_1 — среднее гармоническое `precision` и `recall` при $\beta = 1$.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

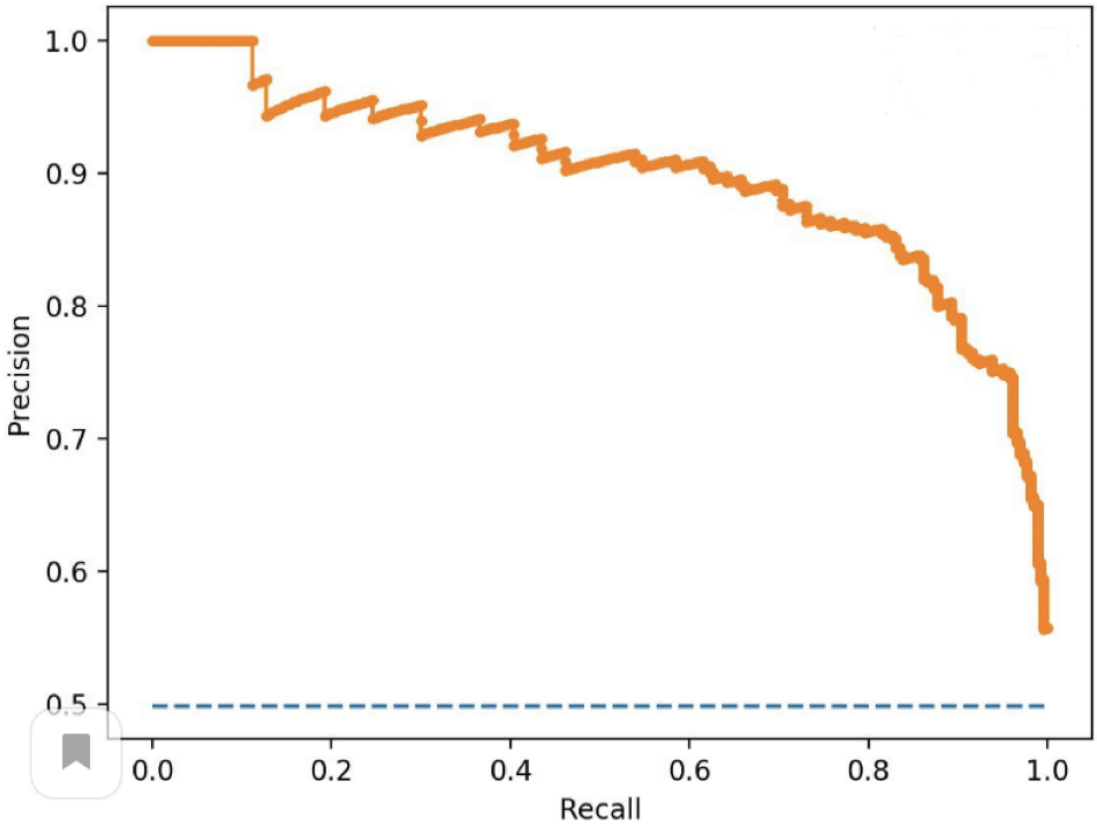
PR-кривая

Построение:

ID оффера	ID модели	Предсказание формулы	Правильный ответ
a01	1	6.4	1
a01	3	0.7	0
b02	2	0.6	1
c03	2	-0.8	0

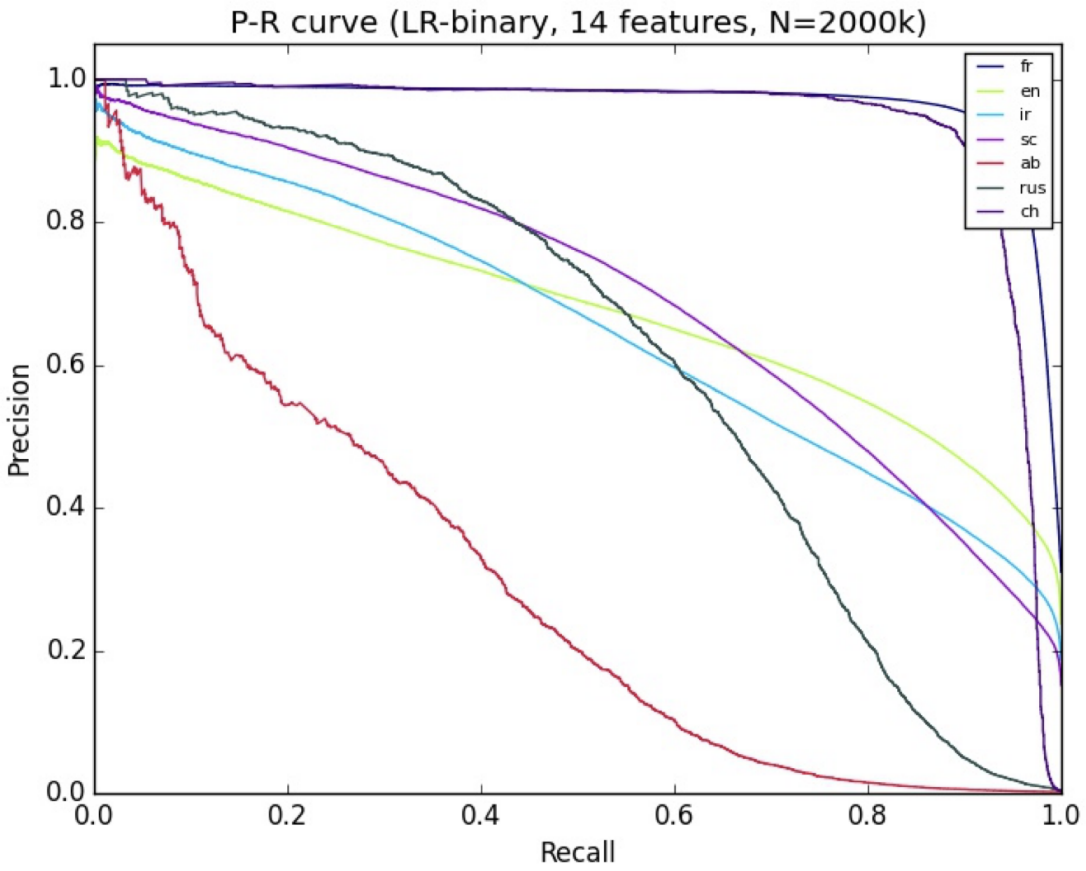
1. Сортируем предсказания по убыванию релевантности.
2. Считаем значение точности и полноты по первой паре.
3. Понижаем значение порога, чтобы выше порога было две пары.
4. Повторяем до тех пор, пока не добавим все элементы.
5. Опционально применить отсечение (Recall@Precision=N).

Метрикой будет площадь под PR-кривой (*PR-AUC*):



PR-кривая

Пример



Рассмотрим несколько графиков `PR` -кривых (представим, что они построены на одних и тех же данных):

Фиолетовая и синяя модели работают практически идеально.

Красная — самая плохая.

Посмотрим на салатовую и бирюзовую линии:

Если мы будем сравнивать модели по `PR-AUC`, то выберем салатовую. Однако если смотреть на первую часть графика, голубая модель лучше при высоких значениях `Precision`.

Поэтому при решении конкретной задачи, возможно, лучше сделать выбор в пользу бирюзовой модели: при той же точности (например, 0.8), значение `Recall` будет выше (больше матчей).

> Average Precision

Average Precision (*AP*) показывает, насколько много релевантных объектов сконцентрировано среди самых высоко оценённых. Чувствительна к порядку ранжирования в топе.

$$AP = \sum_{K}(Recall@k - Recall@[k-1]) \cdot Precision@k$$

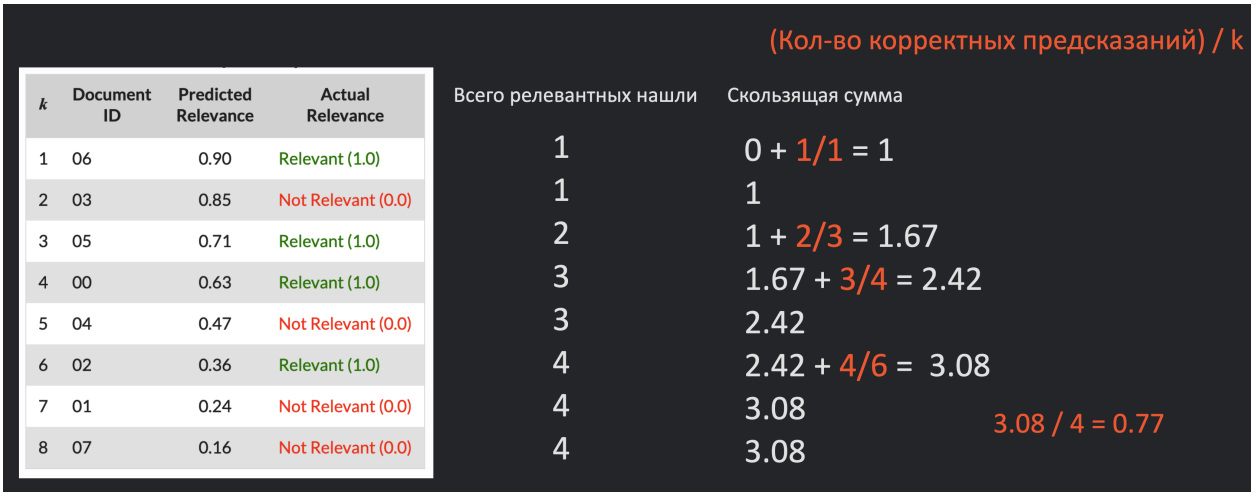
Пример

k	Document ID	Predicted Relevance	Actual Relevance
1	06	0.90	Relevant (1.0)
2	03	0.85	Not Relevant (0.0)
3	05	0.71	Relevant (1.0)
4	00	0.63	Relevant (1.0)
5	04	0.47	Not Relevant (0.0)
6	02	0.36	Relevant (1.0)
7	01	0.24	Not Relevant (0.0)
8	07	0.16	Not Relevant (0.0)

Результаты запроса, отранжированные по предсказанной релевантности (Predicted Relevance)

Наша система нашла 4 релевантных документа — будем считать, что это все релевантные документы.

Теперь посчитаем точность (Precision) до k позиции:



Mean Average Precision

MAP — среднее AP по всем запросам Q

$$MAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q}$$

> Переход к многоуровневой задаче, Gain

Возьмем три уровня релевантности:

1. Нерелевантно;
2. В целом релевантно;
3. Очень релевантно, точное соответствие.

Пример

Есть запрос, для которого найдены 7 документов. Посчитаем функцию Gain, значения которой равны порядковым номерам уровней релевантности.

Потом посчитаем кумулятивную сумму до k объекта. В последней строке "пирамидки" — итоговый кумулятивный Gain (*Cumulative Gain, CG*).

	“Gain”	Cumulative Gain
D1	3	3
D2	2	3+2
D3	1	3+2+1
D4	1	3+2+1+1
D5	3	3+2+1+1+3
D6	1	3+2+1+1+3+1
D7	2	3+2+1+1+3+1+2

Минусы подхода: `CG` нечувствительный к ранжированию внутри набора объектов.

Решение: добавлять штраф за увеличение позиции — чем ниже, тем он больше. Добавляющийся `Gain` будем делить на $log_2(k+1)$, где k — индекс текущей позиции. В итоге получим Discounted Cumulative Gain (*DCG*).

	“Gain”	Discounted Cumulative Gain
D1	3	3
D2	2	3 + 2/log(3)
D3	1	3 + 2/log(3) + 1/log(4)
D4	1	3 + 2/log(3) + 1/log(4) + 1/log(5)
D5	3	…
D6	1	DCG@7 = 3 + 2/log(3) + … + 2/log(8) ~ 7.38
D7	2	
		IdealDCG@7 = 3 + 3/log(3) + … + 1/log(8) ~ 7.83

Для нашего топа DCG@7 = 7.38, а идеальный IdealDCG@7 = 7.83

При полностью правильном ранжировании в примере максимальный `DCG`, который можно получить: $IdealDCG@7 = 7.83$.

Normalized DCG

Получаем ещё одну метрику, которая часто встречается в задачах ранжирования:

$$nDCG@k = \frac{DCG@k}{IdealDCG@k}$$

$nDCG \in [0, 1]$

> PFound (Yandex)

Значение метрики будет оценкой вероятности найти релевантный результат в выдаче модели:

$$pfound = \sum_{i=1}^{n} pLook[i] \cdot pRel[i]$$

$pLook[i]$ — вероятность просмотреть i-й документ из списка

$pRel[i]$ — вероятность того, что i-й документ окажется релевантным (например, $0\%, 50\%, 100\%$ для трёхуровневой шкалы).

Для расчёта $pLook[i]$ используются два предположения:

- результаты ранжирования просматриваются сверху вниз
- процесс прекращается в случае нахождения релевантного результата либо без каких-то определённых причин (например, если "надоело")

$$pLook[i] = pLook[i-1] \cdot (1 - pRel[i-1]) \cdot (1 - pBreak)$$

$pBreak$ — вероятность прекращения просмотра выдачи

> Исторические метрики

MRR

Среднеобратный ранг (Mean reciprocal rank, MRR) — среднее гармоническое между рангами.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

Метрика подразумевает, что есть только один релевантный документ на запрос. Пытаемся оценить, насколько далеко от топа находится этот документ.

Похожа на `AP` с той лишь оговоркой, что релевантный документ один.

Пример

Запрос	Ответы	Правильный ответ	Ранг	Обратный ранг
кочерга	кочерг, кочергей, **кочерёг**	кочерёг	3	1/3
попадья	попадь, **попадей**, попадьёв	попадей	2	1/2
турок	**турок**, турков, турчан	турок	1	1

MRR = (1/3 + 1/2 + 1) / 3 = 11/18 ~ 0.61

Kendall rank correlation coefficient (Kendall's τ)

$$\tau = \frac{(number\ of\ concordant\ pairs) - (number\ of\ discordant\ pairs)}{\binom{n}{2}}$$

$\tau \in [-1, 1]$

`number of concordant pairs` — количество согласованных пар (верно упорядоченных)

`number of discordant pairs` — количество согласованных пар (неверно упорядоченных)

$\binom{n}{2} = \frac{n(n-1)}{2}$ — биномиальный коэффициент

Часто используется в статистике для оценки ранговых корреляций.

> Резюме

- Имеем привилегию отказаться от выдачи;
- Важны только самые-самые первые результаты (1-3);
- Огромный дисбаланс (от нуля до тысяч матчей);
- Финальное решение можно предоставить классификатору;
- Отдельные метрики для разных этапов пайплайна;
- Метрики могут агрегироваться на уровне одного SKU;
- Различие прокси-метрик и бизнес-метрик.